

HOLLAND & KNIGHT LLP

**701 Brickell Avenue
Suite 3000
Miami, Florida 33131
(305) 789-7773**

**Application
for
United States
Letters Patent**

filed on behalf of

Applicant(s): Manoj Khangaonkar
Adwait Sathye

For: System and Method for Integrating
Applications in Different Enterprises
Separated by Firewalls

Attorney Docket: SVL920030058US1

PATENT

**SYSTEM AND METHOD FOR INTEGRATING APPLICATIONS IN DIFFERENT
ENTERPRISES SEPARATED BY FIREWALLS**

CROSS-REFERENCE TO RELATED APPLICATIONS

5 [0001] Not Applicable.

**STATEMENT REGARDING FEDERALLY SPONSORED-RESEARCH OR
DEVELOPMENT**

[0002] Not Applicable.

10

**INCORPORATION BY REFERENCE OF MATERIAL SUBMITTED ON A
COMPACT DISC**

[0003] Not Applicable.

15 **FIELD OF THE INVENTION**

[0004] The invention disclosed broadly relates to the field of information technologies and more particularly relates to the field of business process integration.

BACKGROUND OF THE INVENTION

20 [0005] In the past enterprises have devoted substantial resources to implement custom, standalone information systems that address specific business domain functionality requirements such as accounting, payroll, manufacturing, and distribution. By creating these separate, standalone systems, each individual section of the business process became isolated from the others.

25

Express Mail No. *EV323492730US*

Docket Number SVL920030058US1

PATENT

[0006] Over time, corporate Information Technology (IT) departments have shifted away from in-house development of these custom systems and have attempted to minimize costs by purchasing enterprise applications from various software vendors. Enterprise applications are more generic, providing general business
5 functionality in a pre-packaged product. Typically, enterprise applications include heterogeneous combinations of application systems, hardware platforms, operating systems, third- and fourth-generation languages, databases, network protocols, and management tools. While these applications bring tremendous benefits to the companies that implement them, on an enterprise level, they only exacerbate the
10 proliferation of "process islands" because they are not readily integratable.

[0007] The need for seamless integration of enterprise applications has resulted in the development of various enterprise application integration (EAI) systems. One such EAI system was a hub-and-spoke system developed by
15 CrossWorlds, Inc. (now part of International Business Machines Corporation) that employs a distributed application with agent and server processes sending messages to each other over a network. Further improvements to that system may be required for deployment over a wide-area network (WAN) such as the Internet due to reliability and security issues. One solution is to use HTTP (HyperText Transfer Protocol) as the
20 transport mechanism but further improvement is desirable to enhance security and reliability.

[0008] The Internet has become an important communication medium for business information. The existing infrastructure is far-reaching and its protocol is universally accepted and used. However, a compatibility problem still exists because
25 different nodes in the Internet use different applications programs that use different data structures and different semantics. Moreover, nodes comprising LANs typically

PATENT

use firewalls to separate those LANs from the Internet. Presently communication across enterprise firewalls presents a problem for business process communications among applications in different enterprises. Conventional infrastructures are adequate for business data communication within a LAN but are inadequate for wide area
5 networks. The inadequacy arises from reliability and security concerns. Therefore, there is a need for a business process integration system that provides secure and reliable inter-enterprise communications.

[0009] IBM's MQSeries software is messaging middleware that allows programs to
10 communicate with each other across all IBM platforms, Windows, VMS and a variety of UNIX platforms. It provides a common programming interface (API) to which programs are written. It uses a message queuing approach that provides reliability by storing messages (in a message queue) until the target application is ready to accept the data. Thus, the messages do not have to be resent when for example the host of
15 the target application is not operational. There is a need to extend the operation of messaging middleware across firewalls.

SUMMARY OF THE INVENTION

[0010] A system for integrating applications in different enterprises separated
20 by firewalls comprises: an input for receiving high level business data from a source application; an encryption engine for encrypting the business data to produce encrypted business data; a queue manager for receiving the encrypted business data and for storing the business data for delivery to a target application; and an output for transmitting the encrypted business data to the target application, wherein the system
25 and the target application are separated by at least one firewall.

PATENT

[0011] An application of the invention is realized by practicing a method for integrating applications hosted at different enterprises separated by at least one firewall. The method comprises steps of: receiving data from a source application program; encoding the data according to a message queuing protocol to provide an MQ (message queuing) message; encrypting the MQ message to provide an encrypted MQ message; and transmitting the encrypted MQ message to a destination application program for processing of the data.

[0012] Another application of the invention is realized by a computer readable medium comprising instructions for performing the above steps in a programmable information processing system or apparatus.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 is a block diagram illustration of a business process integration system according to a first embodiment of the present invention.

[0014] FIG. 2 is a block diagram illustration of a business process integration system according to a second embodiment of the present invention.

[0015] FIG. 3 is a high-level block diagram illustrating a system according to the invention.

[0016] FIG. 4 is a flow chart illustrating a method according to the invention.

DETAILED DESCRIPTION

[0017] Referring to FIG. 1, there is shown a block diagram of a business process integration system 100 for integrating applications in different enterprises separated by firewalls according to an embodiment of the invention. The system 100

comprises a first application program 101 residing in a local area network (LAN). An agent 102 couples the first application 101 to a server 103 which acts as a hub for an enterprise application integration system. The agent 102 acts as an interface between the application 101 and the hub server 103 which processes data in a generic format that can be interfaced with other different applications via other agents (not shown). The server 103 interfaces with the first application 101 in a conventional manner. An MQ server (MQ1) 104 is disposed between the server 103 and a firewall 106 that separates the LAN from the Internet.

10 [0018] At the other end of the Internet a second firewall 108 protects a second LAN from actions by other nodes connected to the Internet. The firewall 108 is coupled to second MQ server (MQ2) 110. The MQ2 110 is in turn coupled to a server 115 and to an agent 112. The server 115 can also be used as an application integration hub for other different applications. The agent 112 is coupled to a second application 15 114.

[0019] According to the invention, agent 112 is used for receiving high level business data from a source application such as second application 114 and for transmitting the data for processing by a server (e.g., server 103) separated from the application 114 by the Internet. To ensure security, an encryption engine, possibly integrated into the agent, encrypts the business data to produce encrypted business data. The MQ server 110 acts as a queue manager for receiving the encrypted business data and for storing the business data for delivery to server 103 for processing the data when the target server 103 is ready to process the data.

25

[0020] The firewall 108 is used to filter out or block undesired messages from other nodes connected to the Internet. It can be a single router that filters out unwanted packets or may comprise a combination of routers and servers each performing some type of firewall processing. In this embodiment, the message
5 originating from application 114 is encrypted using the secure sockets layer protocol.

[0021] As the encrypted message traverses the Internet it encounters a first demilitarized zone outside the firewall 108. This DMZ is a middle ground between the trusted internal network on one side of the firewall 108 and the untrusted, external
10 network, such as the Internet in this case, on the other side.

[0022] The encrypted MQ message is then received at the other end of the Internet. At that end the message first encounters a firewall 106 guarding the local area network where the target server 103 is located. The firewall 106 has been programmed to allow passage of the message. The message is then relayed to queue
15 manager 104 that decodes and decrypts the MQ message and passes it to the server 103 for processing. The server 103 is preferably at a hub of a hub-and-spoke middleware messaging system and the agents 102 and 112 are preferably configured as an adapter or spoke in the system. Adapters are written to interface between a generic hub having a well-known application program interface (API) and an
20 enterprise application having a proprietary data structure scheme or semantics.

[0023] As an example, consider the case where the server 103 is hosted at a large enterprise warehouse and application 114 is hosted at a supplier for the warehouse. An order generated by the warehouse may not be compatible with its supplier's enterprise software 114. The middleware described herein integrates the

different applications without the need to adapt one to the other. The use of message queuing provides the reliability of communications required by enterprise applications and the encryption provides the security that enables communication outside of a protected LAN .

5 [0024] Optionally, the agent 112 can be used for bookkeeping purposes to monitor messages being passed between the application 114 and the server 103. For example the agent 112 can send a message to the application 114 to stop sending messages so that it can perform the bookkeeping functions. The agent 112 can also keep a record of the type and number of messages that it processes.

10 [0025] Referring to FIG. 2, a system 200 is substantially similar to the system 100 shown in FIG. 1, except that the MQ message is encrypted according to the HTTPS (HyperText Transport Protocol Secure) protocol. The HTTPS is the protocol for accessing a secure Web server. Using HTTPS in the URL (uniform resource locator) instead of HTTP directs the message to a secure port number rather than the
15 default Web port number of 80. The session is then managed by a security protocol.

 [0026] Using HTTP has the advantage that it can pass the normally available firewalls on Web servers. For more reliable messaging as provided by HTTP, MQ servers 202 and 204 use a reliable message queue system such as MQSeries Internet Passthrough (MQ IPT). MQ IPT also runs on top of the HTTP protocol and can
20 therefore pass through firewalls. However, it also provides all the advantages which MQ messaging brings to applications.

[0027] Referring to FIG. 3, there is shown a high level block diagram illustrating an information processing system 300 according to the invention. The system 300 can be programmed to operate as a server or agent or can host an application to be integrated with other enterprise applications. The system comprises a central processor unit 302, a memory 304, and an I/O subsystem 306. The memory comprises an operating system 312 (e.g., AIX or OS/2) and an application 314 (e.g., applications 102 or 114 of FIG. 1, which can be supply chain management, order fulfillment or other enterprise software). The system 300 further comprises a CD ROM or DVD drive 308 for receiving a CD ROM 310. The CD ROM 310 may comprise a program product comprising instructions for carrying out methods according to the invention. The CD ROM 310 preferably comprises a hub such as an interchange server and a plurality of adapters each for interfacing with a specific enterprise application. Alternatively, the information processing system 300 may comprise an application specific integrated circuit (ASIC) hardwired to operate according to an embodiment of the invention or a read-only memory may comprise the program instructions required to practice the invention.

[0028] Referring to FIG. 4, there is shown a flow chart illustrating an information processing method 400 according to an embodiment of the invention. The method 400 comprises the following basic acts. In step 402 a remote agent or other information processing system according to the invention receives a message from an application 114. The message comprises high level data and a request to process the data by a server. In step 404 the system converts the message into an MQ message using a message queuing protocol. In step 406 the MQ message is encrypted using a security protocol to provide a secure MQ message. In decision 408 it is determined

whether the packets of the message can be received by the target or destination node. If the target is ready to receive the packets the process continues at step 410. If the target is not ready then the message is stored until the target is ready to accept the message. Finally, in step 410 the MQ message is sent to a first queue manager for
5 retransmission at a time when the network is ready for transporting the message to the target node.

[0029] Therefore, while there has been described what is presently considered to be the preferred embodiment, it will be understood by those skilled in the art that
10 other modifications can be made within the spirit of the invention.

[0030] We claim: